TheJackFolio

This document will give you complete understanding of my portfolio name called “TheJackFolio”. It not only gives information regarding my portfolio but also gives a detail technical aspect of things that I did in the background, issues that I faced while working on this project and many more. So, let’s get started.

1. What is Portfolio Website?
2. Why we need a Portfolio Website?
3. Things to show in TheJackFolio
4. Tech Stack for this project
5. Repository links and complete description
6. External APIs used for this project
7. **What is Portfolio Website?**

A portfolio website is a unique way to showcase your work and let others know about yourself. It's like an evergreen platform for your projects, case studies, and information about you. In addition, it's one of the best ways to express your personality, experience, and capabilities. Much like your PDF portfolio, a portfolio website can be used when applying for jobs or internships.

Portfolio websites come in all shapes and sizes, and can vary depending on your discipline. Ultimately, whether you’re a designer, photographer, writer or digital artist, the best portfolio websites are a true reflection of your projects, passions and personality.

1. **Why we need a Portfolio Website?**

Today, there are loads of ways to showcase your work online, be it through an Instagram account, GitHub account, LinkedIn profile. And depending on your discipline or practice, showcasing your work on a particular platform (or combination of platforms) might make the most sense for you.

There are a few reasons, however, why it’s worth having a website, too:

* **Control over your online presence**

Having a portfolio website gives you the opportunity to control and curate the way you want to be seen online, in one dedicated space.

Whereas other online platforms might limit you to certain dimensions or formats, having your own online space gives you the flexibility and freedom to choose everything from the layout and the size of images to the typefaces you use.

* **Help people find you online**

Without one, clear destination for your work, potential employers might have to click around on various platforms. Your portfolio website, however, can help people find you faster online by including the right search terms and tags (more on that later).

This way, your website will usually be the first thing to appear in search results when people look you up. Plus, this can minimize confusion if, for example, you have a common name.

* **A reliable way to host your work**

Social media platforms offer an easy and accessible way to showcase your work, but not without certain risks. A portfolio site can serve as an effective backup for any work you share on social media, in the event that your account gets banned or suspended for any reason – even hacked. It happens more commonly than you think, and often when you least expect it.

You also won’t have to worry about the risk of platforms updating overnight and having to navigate through new interfaces!

* **Website’s purpose**

Before you start collecting projects or drawing out your dream website design, it’s a good idea to think about what you want to achieve with your portfolio website. For example, do you want your website to be:

A one-stop destination for all your creative work

A space to flex your coding and design skills

A way to point people to your blog, shop or social media accounts

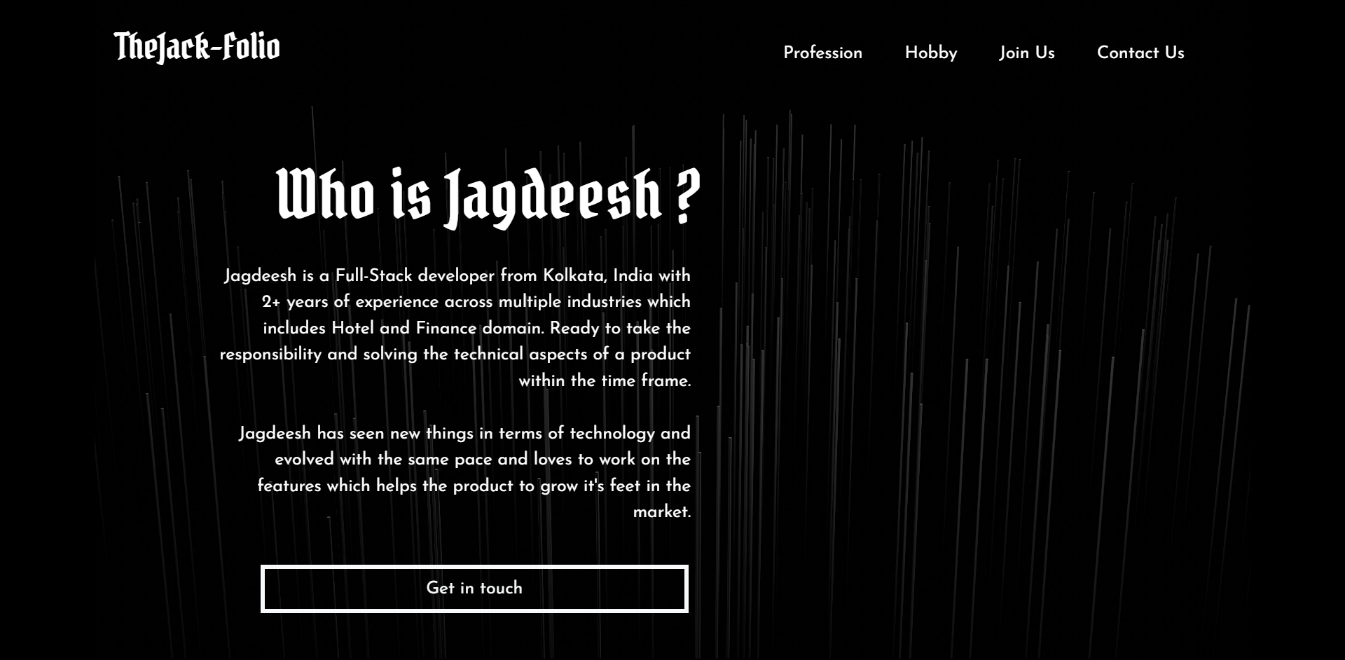
The answer to this will be different for everyone, it could even be a combination of the above. Whatever the answer, knowing what role you want your website to play will help make the decision-making process much easier. For my website both 1st and 2nd can be applicable.

1. **Things to show in TheJackFolio**

I designed my Portfolio Website in such a way that I can show both my Professional and Personal Journey till now. Starting from the organizations that I worked for till the creative work that I post in different social medias can be find here.

Website Link:

* Home Page



1. **Tech Stack used for this Portfolio**

* Backend Framework: Spring Boot@3.1.2
* Frontend Framework: React JS@18.2.0
* Database: MySQL
* Used Microservice Architecture

1. **Repository links and complete descriptions**

There are several projects that are being created and used for this Portfolio. We will go through each of them and understand both business and technical aspect of the projects.

* **microservice-thejackfolio\_db**

Repository Link: https://github.com/Xaptured/microservice-thejackfolio\_db

Description: This microservice is used as a layer on the Database to do all kind of operations like add, update, select and delete. There are other microservices which are using this service to get data from the database or save data in the database.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Step 3: Please select the appropriate environment like dev or prod in “application.properties” which internally selects the appropriate profile and change the database values accordingly.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

H2-Console: http://localhost:{your-port}/h2-console

* **microservice-client**

Repository Link: https://github.com/Xaptured/microservice-client

Description: This microservice is used to get the client related information like comments and details which are being processed and sent to database layer(microservice-thejackfolio\_db).

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

* **microservice-profession**

Repository Link: https://github.com/Xaptured/microservice-profession

Description: This microservice is used to initialize all the of my professional and personal information in the database and send them to the website.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

* **microservice-youtube**

Repository Link: https://github.com/Xaptured/microservice-youtube

Description: This microservice is used as a layer above the YouTube APIs which can do operations related to channel data, media information etc.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

Please go to the 6th section to know more about YouTube APIs.

* **microservice-instagram**

Repository Link: https://github.com/Xaptured/microservice-instagram

Description: This microservice is used as a layer above the Instagram Basic Display APIs which can do operations related to media data, posts information etc. Along with this there is a scheduler which will run everyday at 1:15 PM to check whether the long lived access token is going to expire or not in the coming 2 days.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to keys.properties and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

Please go to the 6th section to know more about Instagram Basic Display APIs.

* **microservice-instagram-oauth-client**

Repository Link: https://github.com/Xaptured/microservice-instagram-oauth-client

Description: This microservice is used as a layer above the Instagram Authorization APIs which are used to do authorization and get access token.

Please keep in mind that this service is a https protected so all the URLs which access the resources in this service will be https enabled.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path src/main/java/resources and there you will find keys\_dummy.properties file.

Step 2: Rename the file from keys\_dummy.properties to keys.properties and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Other links that can be useful while running this repo in local:

Swagger UI: http://localhost:{your-port}/swagger-ui/index.html

Please go to the 6th section to know more about Instagram Authorization APIs.

* **microservice-registry**

Repository Link: https://github.com/Xaptured/microservice-registry

Description: This microservice is used as a Service Discovery for all the services which are required for this project.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

* **microservice-api-gateway**

Repository Link: https://github.com/Xaptured/microservice-api-gateway

Description: This microservice is used as an API Gateway for the other services. All the requests which are coming from frontend side will hit this service and this service is responsible to route those requests to the required services.

If you want to clone this repository and use it in your local then please follow these steps:

Step 1: Please go to the path “src/main/java/resources” and there you will find “keys\_dummy.properties” file.

Step 2: Rename the file from “keys\_dummy.properties” to “keys.properties” and add your values accordingly. You can also take reference from Google to fill the appropriate keys in the file.

Note: All the actuator end points are protected with password. Please configure accordingly to use them.

* **thejackfolio\_dashboard**

Repository Link: https://github.com/Xaptured/thejackfolio\_dashboard

Description: This is the actual dashboard for my portfolio website.

* **TheJackFolioParentRepository**

Repository Link: https://github.com/Xaptured/TheJackFolioParentRepository

Description: This repository is to track all the work done for this project.

Under Issues you can see 6 epic links which contains all the tickets that are being created for this project.

Each ticket will be having a PR which is being used to close the ticket.

**Project Dashboard Link:** https://github.com/users/Xaptured/projects/2

Please go through the tickets under TheJackFolio Project (link mentioned above) if interested to know more about the work flow that I followed to complete this project.

1. **External APIs used for this project**

There are mainly 2 external APIs that I used to create this project.

* Google APIs to access YouTube media data
* Instagram APIs

I had to go through lots of articles to understand the complete process through which we can access these APIs. So, I thought to document them in a single place so that whoever go through this document can understand more about this process.

* **Google APIs**

Step-1: Make sure you are logged in to the correct Google account.

Step-2: Open <https://console.cloud.google.com/>

Step-3: Create a new project which you can find side to the Google Cloud logo.

Step-4: Choose the newly created project and navigate to APIs & Services tab

Step-5: Search for YouTube Data API v3 and enable it.

Step-6: Navigate to credentials and select “create credentials”

Step-7: Under dropdown select API Key and booooom

You will get your API key which can be used to get the data from Google.

Note: This API Key doesn’t have an expiration date until and unless you delete this key from your Google Cloud Console Project.

* **Instagram APIs**

Step-1: Make sure you are logged into the correct Facebook profile.

Step-2: Open <https://developers.facebook.com>

Step-3: Click on My Apps

Step-4: Create a new app and select for everything else

Step-5: Add necessary data and click Create App ID

Step-6: Navigate to settings and click on Basic

Step-7: Scroll down to the end and click on Add Platform then Website and add your website URL

Step-8: Click on Products from left side and search for Instagram and click on Set Up

Step-9: Click on Basic Display and create new App with display name

Step-10: Enter Valid OAuth Redirect URLs and save

Step-11: Then find add or remove test users and go to Instagram Testers section where you need to add your instagram profile ID which will show status as pending.

Step-12: To Confirm this status we need to move to your instagram account and go to settings then open Apps and Websites. Under tester invites you will see one request. Accept that.

Step-13: Now again go to the Facebook developer page and navigate to Basic Display. Under User Token Generator you will see your instagram profile with button called generate token.

Note: The secret that you get from developer portal can be used to get access token. The complete process is described below.

Step-1: Get authorization code

Step-2: Get the access token using auth\_code (expiration time – 1 hour)

Step-3: Get Long Lived Access Token with help of access\_token (expiration time – 60 days)

Step-4: Get Refresh Token with help of long\_lived\_access\_token (should be used before using expiration of long\_lived\_access\_token)

Step-5: This long\_lived\_access\_token can be used to get the media data and information from Instagram Tester Account.